Journal of Modern Education Review, ISSN 2155-7993, USA
December 2024, Volume 14, No. 11–12, pp. 371–383
Doi: 10.15341/jmer(2155-7993)/ 11–12.14.2024/001
© Academic Star Publishing Company, 2024
http://www.academicstar.us

A New Gamified Method for Discovering and Understanding the Conversion From Binary to Decimal Numbers and Their Representation on Seven-Segment Displays, Using FPGAs and VHDL, for Student Application

Evangelos I. Dimitriadis[1], *L. Dimitriadis*[2], *T. Dimitriadou*[3]

(1. Department of Computer, Informatics and Telecommunications Engineering, International Hellenic University, Serres, Greece;
2. Department of Information and Electronic Engineering, International Hellenic University, Thessaloniki, Greece;
3. Aristotle University of Thessaloniki, Greece)

Abstract: A new method which uses a gamified and also experimental procedure for investigation, discovering and understanding the conversion from binary to decimal numbers, and the way of representing them in seven-segment displays, is presented here for the first time. In order to achieve the goal, FPGAs and the hardware programing language VHDL are used. Students also gain experience on a modern hardware programing language and they obtain applicable results from their program, thus enhancing their algorithmic thinking. In practice, two quads of FPGA's input switches are used, one for each student, and depending on which switches they set to ON or OFF, they try to display the requested decimal number on the seven-segment display. If both answers are correct, all the LEDs light up. Otherwise, the LEDs are off and the characters FA (FALSE) are displayed so they must try to form the correct number again. In all cases, the two numbers formed from both students appear in the displays.

Key words: gamified learning method, binary to decimal, FPGAs, VHDL, seven-segment display representation

1. Introduction

It is widely known (Piaget Jean et al., 1952; Bovet Magali, 1976; Cherry K., 2020; Kurt S., 2020; Bruner J. S., 1978, 1983, 1995; Gagne R. M., 2005; Erotokritou Stavrou Th., & Koutselini, M., 2016; Jewitt C., 2008) that active teaching methods, in which student investigates, experiments and ultimately discovers knowledge experientially, result in better knowledge consolidation. In fact, if the whole process uses STEM, collaborative and project learning methods (Artigue M., 2006; Avouris N., Dimitracopoulou A., & Komis V., 2003; Daher W., &

Evangelos I. Dimitriadis, Ph.D., Research Associate, Department of Computer, Informatics and Telecommunications Engineering, International Hellenic University; research areas: micro/nano electronics-simulation, modeling and device fabrication, FPGAs programing and applications. E-mail: edimitriad@sch.gr.
L. Dimitriadis, Undergraduate Student, Department of Information and Electronic Engineering, International Hellenic University.
T. Dimitriadou, MSc, Aristotle University of Thessaloniki; research areas: English language and Literature.

Awawdeh Shahbari J., 2020; Kennedy T. J., & Odell M. R. L., 2014; Handrianto C., & Rahman M. A., 2018; Faber J. M., Glas C. A. W., & Visscher A. J., 2018) and contains a gamified way of learning and finally presents the acquired knowledge to the rest of the trainees, then the results are even better.

Another interesting fact is that Field Programmable Gate Arrays (FPGAs), have attracted research interest for industrial, as well as other applications (Yussup M., 2014; Gujar A., 2014; Singh S., Saini A. K., & Saini R., 2014; Muthukrishnan S., & Priyadharsini R., 2014; Chen L., Chang Y., & Yan L., 2022; Yarlagadda S. et al., 2021; Du C., & Yamaguchi Y., 2020; Hao X., Lin C., & Wu Q., 2020; Bawiskar P. A., & Agrawal R. K., 2015; Saroch K., Sharma A., 2013; Parikh, R. S. 2018; Sebastian Strube, Gabor Molnar, & Hans-Ulrich Danzebrink, 2011; Sanjay Singh, Chandra Shekhar, & Anil Vohra, 2017; Joel Dunham, & Eric Johnson, 2012), but there is a lack of FPGAs' educational applications. Our work combines new teaching methods with a technologically advanced and pioneer area, in order to achieve better educational results.

Computer simulations for understanding binary-to-decimal conversion are useful, but our method using FPGAs and on-board switches and displaying the results on seven-segment displays, has significant advantages. The student acts on his own and discovers knowledge, handling hardware and its behavior and additionally manages to experience the use of a hardware programming language in this kind of applications. He also discovers in practice, the use of algorithmic structures. Our method is aimed at teenage students, who do not yet have a clear knowledge of the binary system. It can also be used by teachers of other specialties and not only Informatics teachers. It is also possible to use our method for university students, where it could be extended so that more than 4 switches representing respective bits could be used to represent bigger decimal numbers.

The results of applying our method in the classroom were very encouraging and certainly achieved the original goal to a large extent.

2. Software and Hardware Used in This Work

VHDL is the hardware programing language used here and Intel's Quartus Prime is the programing tool for creating VHDL files. VHDL is a hardware description language (Hardware Description Language), which is used to develop integrated digital circuits and systems. VHDL code is the basic file type accepted as input to digital circuit design software (Computer Aided Design or CAD), to create complex integrated circuits. The VHDL language is widely used to describe and implement digital systems in FPGAs (Field Programmable Gate Arrays).

VHDL differs from conventional languages because it is not intended to describe operations performed serially, one after the other. Each sentence or section code describes functions, which produce results in synchronization with other parallel functions. The simulation results are produced based on strict time specifications at various points of the circuit and finally at the outputs of the FPGA.

Field Programmable Gate Arrays (FPGAs) are digital circuits consisting of a large number of logical gates which are electrically connected via semiconductor switches. VHDL program activates the FPGA's circuits needed each time in order to exhibit the appropriate circuit behavior (counter, multiplexer, ALU, memory etc.).

We used the DE10 Lite FPGA board with an Altera MAX10 generation chip, shown in Figure 1.

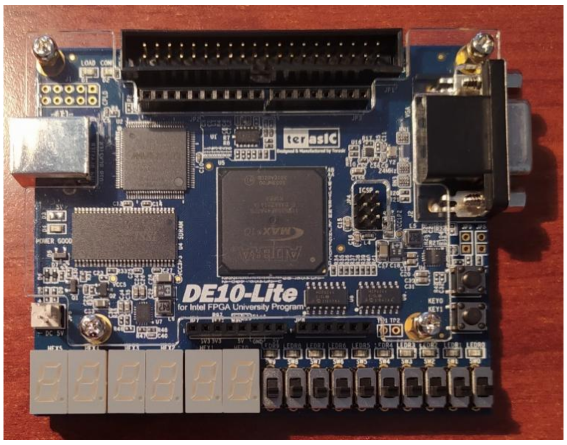

Figure 1 DE10 Lite FPGA Board Used in This Work

3. FPGA Programing and Operation

The main function of the VHDL program used here is shown in the block diagram of Figure 2.

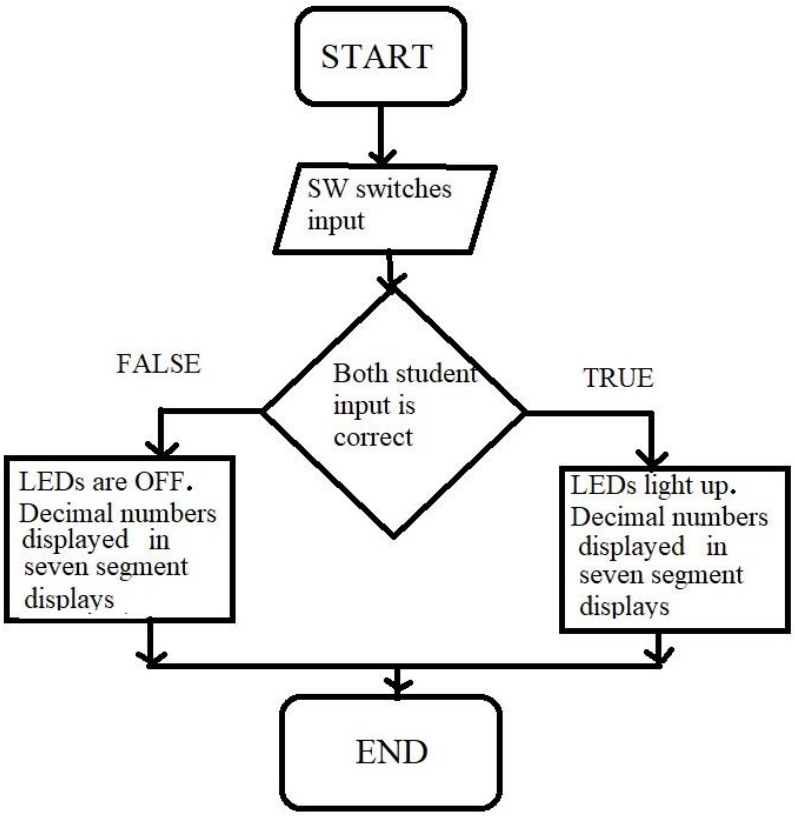
START
SW switches input
Both student input is correct
FALSE
TRUE
LEDs are OFF. Decimal numbers displayed in seven segment displays
LEDs light up. Decimal numbers displayed in seven segment displays
END

Figure 2 Block Diagram of the VHDL Program Used in Our Method

Both students use four different to each other FPGA board switches, attempting to represent correctly the requested decimal number. The two inputs are combined and if both students are correct all LEDs light up, while at the opposite case LEDs are OFF. Decimal numbers formed from both students are represented at seven-segment displays.

The program consists of three parts. The first one is the switches input part. The second is the condition check part and third one is LEDs and seven-segment displays' behavior.

A VHDL program always starts with entity and continues and finishes with architecture. The above block diagram of Figure 2 corresponds to architecture part. Entity follows libraries' declarations and contains all ports (IN, OUT, buffers etc.) used in the program. Below is entity used in our program:

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY Project_game    IS
PORT(c :IN std_logic_vector (3 DOWNTO 0);
d :IN std_logic_vector (3 DOWNTO 0);
e :buffer std_logic_vector (3 DOWNTO 0);
ex1: OUT std_logic_vector(6 DOWNTO 0);
ex2: OUT std_logic_vector(6 DOWNTO 0);
ex3: OUT std_logic_vector(6 DOWNTO 0);
ex4: OUT std_logic_vector(6 DOWNTO 0);
ex5: OUT std_logic_vector(6 DOWNTO 0);
ex6: OUT std_logic_vector(6 DOWNTO 0);
led1: out std_logic;
led2: out std_logic;
led3: out std_logic;
led4: out std_logic;
led5: out std_logic;
led6: out std_logic;
led7: out std_logic;
led8: out std_logic;
led9: out std_logic;
led10: out std_logic
);
END Project_game;
```

It is obvious that c and d inputs correspond to the two quads of FPGA's switches, while ex1, ex2, ex3, ex4, ex5 and ex6 are the outputs of seven-segment displays. FPGA's LEDs also act as output.

The architecture part of our program, as mentioned above, corresponds to the block diagram of Figure 2, except for switches' input, which comes from entity part. Checking of the inputs follows and is shown below using a process, where the signals state_LED and state_e are related to LEDs lighting and e output to seven-segment display, respectively:

```
process (c,d) begin
  IF (c=d)    AND (c="1001") then-- we set 1001 as the right number
    state_LED <= '1';
    state_e <="0001";
```

```
 else
   state_LED <= '0';
   state_e <="0000";
   end if;
 end process;
```

Finally the LEDs lighting and seven-segment display part of our program are shown below:

```
ARCHITECTURE behaviour OF Project_game IS
signal state_LED: std_logic;
signal state_e: std_logic_vector (3 DOWNTO 0);
BEGIN


WITH c SELECT
ex1<= "1000000" when "0000", -- 0, active low, i.e., 0:display & 1:no display
        "1111001" when "0001", -- 1
        "0100100" when "0010", -- 2
        "0110000" when "0011", -- 3
        "0011001" when "0100", -- 4
        "0010010" when "0101", -- 5
        "0000010" when "0110", -- 6
        "1111000" when "0111", -- 7
        "0000000" when "1000", -- 8
        "0010000" when "1001", -- 9
        "1000000" when "1010", --10
        "1111001" when "1011", --11
        "0100100" when "1100", --12
        "0110000" when "1101", --13
        "0011001" when "1110", --14
        "0010010" when "1111"; --15

WITH c SELECT
ex2<= "1000000" when "0000", -- 0, active low i.e. 0:display & 1:no display
        "1000000" when "0001", -- 1
        "1000000" when "0010", -- 2
        "1000000" when "0011", -- 3
        "1000000" when "0100", -- 4
        "1000000" when "0101", -- 5
        "1000000" when "0110", -- 6
        "1000000" when "0111", -- 7
        "1000000" when "1000", -- 8
        "1000000" when "1001", -- 9
        "1111001" when "1010", --10
        "1111001" when "1011", --11
        "1111001" when "1100", --12
        "1111001" when "1101", --13
        "1111001" when "1110", --14
        "1111001" when "1111"; --15

WITH d SELECT
ex3<= "1000000" when "0000", -- 0, active low i.e. 0:display & 1:no display
        "1111001" when "0001", -- 1
        "0100100" when "0010", -- 2
        "0110000" when "0011", -- 3
        "0011001" when "0100", -- 4
        "0010010" when "0101", -- 5
        "0000010" when "0110", -- 6
        "1111000" when "0111", -- 7
        "0000000" when "1000", -- 8
        "0010000" when "1001", -- 9
        "1000000" when "1010", --10
        "1111001" when "1011", --11
        "0100100" when "1100", --12
        "0110000" when "1101", --13
        "0011001" when "1110", --14
        "0010010" when "1111"; --15

WITH d SELECT
ex4<= "1000000" when "0000", -- 0, active low, i.e., 0:display & 1:no display
        "1000000" when "0001", -- 1
        "1000000" when "0010", -- 2
        "1000000" when "0011", -- 3
        "1000000" when "0100", -- 4
        "1000000" when "0101", -- 5
        "1000000" when "0110", -- 6
        "1000000" when "0111", -- 7
        "1000000" when "1000", -- 8
        "1000000" when "1001", -- 9
        "1111001" when "1010", --10
        "1111001" when "1011", --11
        "1111001" when "1100", --12
        "1111001" when "1101", --13
        "1111001" when "1110", --14
        "1111001" when "1111"; --15
led1 <= state_LED;
led2 <= state_LED;
led3 <= state_LED;
led4 <= state_LED;
led5 <= state_LED;
led6 <= state_LED;
led7 <= state_LED;
led8 <= state_LED;
led9 <= state_LED;
led10 <= state_LED;
e <= state_e;

WITH e SELECT
ex5<= "0111111" when "0001",
        "0001110" when "0000",
         "1111111" when others;

WITH e SELECT
ex6<= "0111111" when "0001",
        "0001000" when "0000",
        "1111111" when others;

END behaviour;
```

It was mentioned above that outputs ex1, ex2, ex3, ex4, ex5 and ex6 are related to seven-segment displays.

As shown in Figure 1, DE10 Lite FPGA board is equipped with six seven-segment displays, appearing at the down left part of the board. Outputs ex1, ex2, ex3, ex4, ex5 and ex6 correspond to board's displays from left to right 2, 1, 5, 6, 4 and 3, respectively, as shown in Table 1.

Table 1 Outputs ex1, ex2, ex3, ex4, ex5 and ex6 Correspond to Board's Displays

Display 1	Display 2	Display 3	Display 4	Display 5	Display 6
ex2	ex1	ex5	ex6	ex4	ex3
tens	units	— or F	— or A	tens	units

It is obvious that the first two-digit number created from student1 using four FPGA's SW switches, is shown in display1 and display 2, while the second number created from the other student is shown in display 5 and display 6. Displays 3 and 4 show when both numbers are correct and F A when at least one number is wrong.

All the above are presented in Figures 3 and 4.

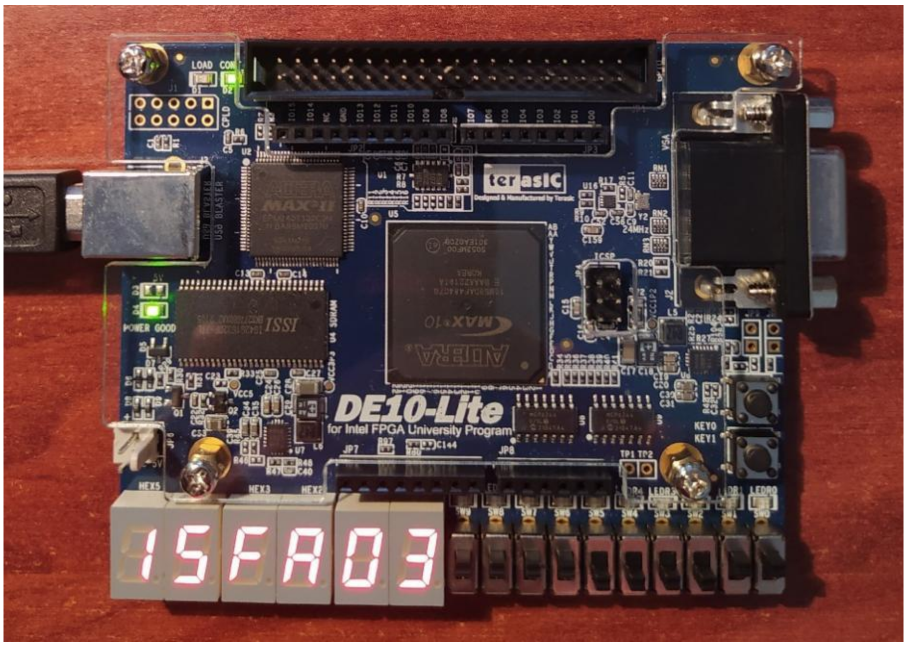

Figure 3 Seven-Segment Displays' Results When One or Both Students Formed Wrong Number

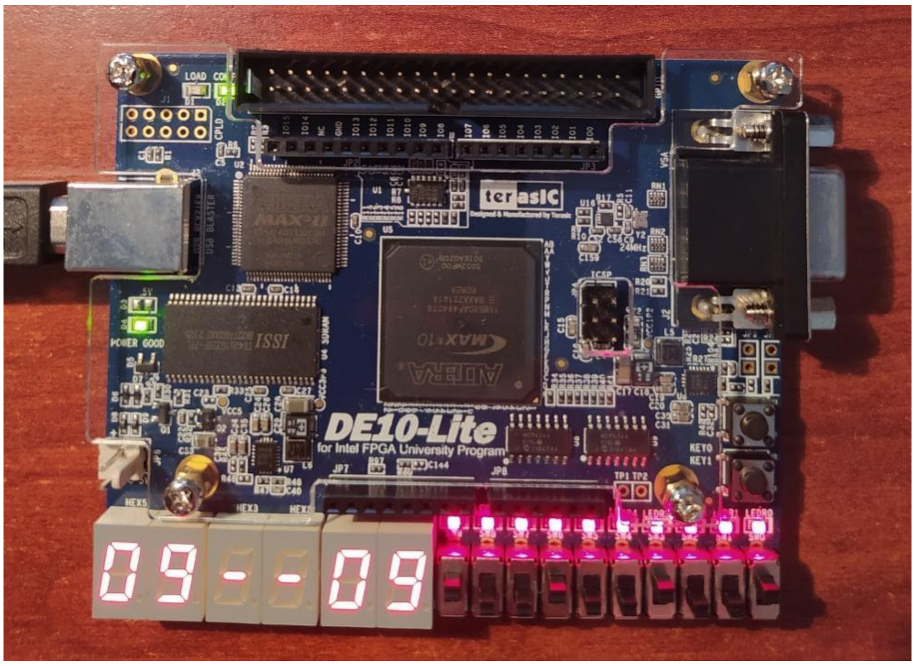

Figure 4 Seven-Segment Displays' Results When Both Students Formed the Correct Number

Figure 3 presents the results of wrong number formation, while Figure4 shows the results when both students form the right number, using SW switches.

Another very important procedure that needs to be well explained to students, is the way of presenting in seven-segment displays, the decimal numbers resulting from the conversion of corresponding binary numbers. Since the conversion is done in a way that will be explained later, next step is the part of VHDL code that drives seven-segment displays to present the decimal number. It is worth noting here that each imaging unit (digital tube) contains seven sectors (seven-segment display) and each of them must be commanded to light up or not, in order to form one digit of the decimal number.

Notice that seven segments start at A and end at G, as shown in Figure 5.

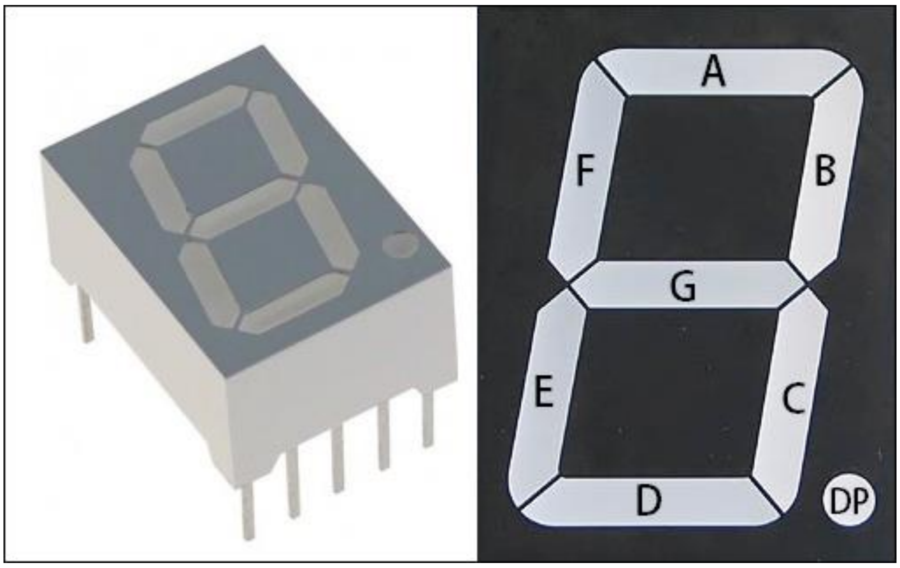

Figure 5 Seven-Segment Display Digital Tube

The De10 Lite FPGA board of this work, uses active low method at its seven-segment displays. This means that every segment is activated so it lights up, when it receives a zero (0) bit and it is OFF when it receives a one (1) bit. Taking into account all the above facts if, for example, number 3 needs to be represented then “0110000” must be sent to a digital tube where from right to left, A segment receives 0, B receives 0, C receives 0, D receives 0, E receives 1, F receives 1 and G receives 0. All segments receiving 0 will light up and the opposite holds true for those which received 1, thus forming number 3.

Based on the above mechanism the programmer must prepare an appropriate program for succeeding correct decimal numbers representation. The use of multiple choice algorithmic structure is used for all outputs related to seven-segment displays of this work. An example for ex1 is shown again for better understanding and it will be explained to students in detail:

```
WITH c SELECT
ex1<= "1000000" when "0000", -- 0, active low, i.e., 0: display & 1: no display
      "1111001" when "0001", -- 1
      "0100100" when "0010", -- 2
      "0110000" when "0011", -- 3
      "0011001" when "0100", -- 4
      "0010010" when "0101", -- 5
      "0000010" when "0110", -- 6
      "1111000" when "0111", -- 7
      "0000000" when "1000", -- 8
      "0010000" when "1001", -- 9
```

```
"1000000" when "1010", --10
"1111001" when "1011", --11
"0100100" when "1100", --12
"0110000" when "1101", --13
"0011001" when "1110", --14
"0010010" when "1111"; --15
```

4. Applying Our Method in Students' Classroom

We applied our method to students, during four non-consecutive teaching hours.

1st hour:

At first students become familiar with FPGAs and VHDL. They are organized into groups of 3–4 people and each group watches an informational video using a personal computer and if needed, having teacher's guidance and explanations. Working in their groups, they try to discover answers to a variety of simple questions concerning FPGAs and VHDL. Then each group makes a brief presentation about the most important concepts of FPGAs and VHDL and the answers they gave. Next comes a short demonstration by the professor about Intel's Quartus VHDL programming environment using a video projector. Finally in the last few minutes of the teaching hour, brainstorming takes place, concerning Quartus and the students' impressions about it, compared to other programing environments.

Needless to say that all the above of 1st teaching hour's work, is not necessary for students familiar with FPGAs, VHDL and Quartus. They can spend the hour programing the FPGA.

2nd hour:

Now is the time of binary to decimal conversion study. Students are watching a few minutes presentation from teacher concerning the above issue. The main aspects of the presentation are given below.

The binary number can be converted to a decimal number by expressing each digit as a product of the given number 1 or 0 to the respective power of 2.

The decimal number for a given binary one of the form $(a_5\ a_4\ a_3\ a_2\ a_1\ a_0)_2$ is, $D = (a_5\times2^5) + (a_4\times2^4) + (a_3\times2^3) + (a_2\times2^2) + (a_1\times2^1) + (a_0\times2^0)$.

We can convert 10101 to the decimal number form in the following way:

The binary number 10101 is expressed as $(10101)_2 = (1\times2^4) + (0\times2^3) + (1\times2^2) + (0\times2^1) + (1\times2^0) = (21)_{10}$. Thus, the binary number 10101 is expressed as $(21)_{10}$.

Table 2 and Table 3 present powers of 10 and powers of 2, respectively. The third row of Table 3 has values of 1 or 0, at the appropriate positions, in order to form (10101) of binary system.

It should be mentioned here that each one of four FPGA switches used by each student, correspond to a power of 2, as shown in Figure 6. Thus when the switch is ON means that the specific power of 2 has a factor of binary 1. Factor 0 goes to that power of 2 if switch is OFF.

Table 2 Decimal System

10000	1000	100	10	1
10^4	10^3	10^2	10^1	10^0

Table 3 Binary System

512	256	128	64	32	16	8	4	2	1
2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	0	0	0	1	0	1	0	1

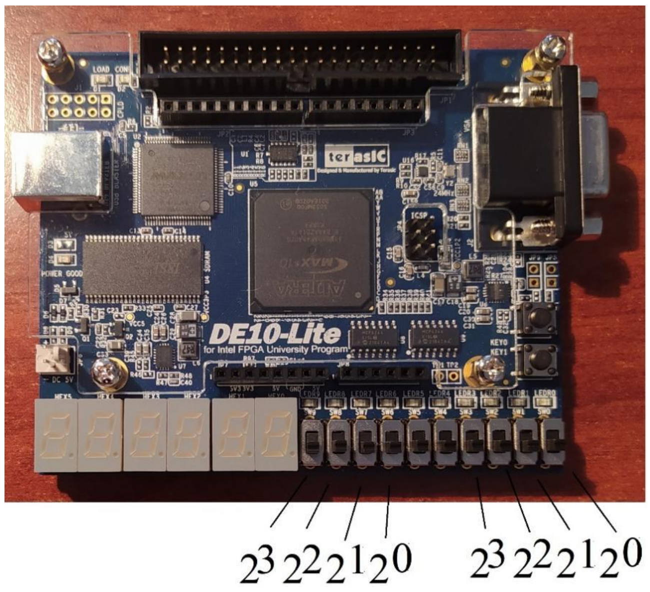

Figure 6 Powers of 2 Corresponding to FPGA's Switches Used by Two Students

Since we are using four switches for each student, it is obvious that we are using four bits for binary numbers, thus the maximum decimal number which can be represented is 15. All decimal numbers from 0 to 15 can be represented in seven-segment displays.

Study continues with lots of examples given by the teacher, needed to be solved from students.

Finally students are experiencing the process for representing the decimal number that came out of the conversion, in seven-segment displays with the way it was presented in previous part 3 of this work. Understanding the above way of representation is crucial for students because it will be used in the program they will write below.

3rd hour:

The main goal was to successfully program the FPGA in VHDL language, in order to perform our method correctly. Block diagram of Figure 2 is given to students, along with stating the problem we are trying to solve. Students are divided to groups and each group provides the part of the program needed to form total program.

4th hour:

As a culmination of the effort made by the students, the most pleasant hour of gamified learning, follows.

A practical implementation of representing decimal numbers on the FPGA's seven-segment displays, takes place. Students play multiple roles. Some of them ask for specific decimal number formation, others solve the problem in their papers and other students try to form the right decimal number, using FPGA's switches. All students manage to alternatively play all the roles.

In conclusion, we could say that we achieved the desired goal to a large extent.

As it was expected, there was a difficulty when writing the program, since the students' experience with VHDL was relatively small. Students showed great interest and had fun, when acting in groups and playing the role of presenter, demonstrating to their classmates, using FPGA board switches, the way of practically representing in seven-segment displays, specific numbers, which other students asked for.

5. Conclusions

We presented here, for the first time, a new method for investigating, discovering and understanding the conversion from binary to decimal numbers, and the way of representing them in seven-segment displays. Our method uses a gamified and also experimental procedure in conjunction with the use of FPGAs and the hardware programing language VHDL. Students work in groups and present their results from gained experience on a modern hardware programming language (VHDL) and its applicable results, thus enhancing their algorithmic thinking. They also practice with the way of converting binary to decimal numbers and representing decimal numbers to seven-segment displays, by using appropriate algorithmic structures. Students finally use two quads of FPGA's input switches, one for each student, and depending on which switches they set to ON or OFF, they try to display the requested decimal number on the seven-segment display. All FPGA's LEDs light up only if both students' answers are correct. Otherwise, the LEDs are off and the characters FA (FALSE) are displayed and students must try again to form the correct number. In all cases, the two numbers formed from both students appear in the displays. Students play multiple roles. Some of them ask for specific decimal number formation, while other students try to form the right decimal number, using FPGA's switches. In conclusion, we could say that the desired goal was sufficiently achieved.

References

Artigue M., Kynigos C., Mariotti A., Cerulli M., Lagrange J. B., Bottino R. M., Haspekian M. and Cazes C. (2006). "Methodological tools for comparison of learning theories in technology enhanced learning in mathematics", interim report of the Kaleidoscope European Research Team 'Technology Enhanced Learning of Mathematics, available online at: http://www.itd.cnr.it/telma.

Avouris N., Dimitracopoulou A. and Komis V. (March 2003). "On analysis of collaborative problem solving: An object-oriented approach", *Computers in Human Behavior*, Vol. 19, No. 2, pp. 147–167.

Bawiskar P. A. and Agrawal R. K. (Dec. 2015). "FPGA based home security system", *International Journal of Innovative Research in Science, Engineering and Technology*, Vol. 4, No. 12, pp. 12865–12869, doi: 10.15680/IJIRSET.2015.0412139.

Boring Edwin G., Werner Heinz, Langfeld Herbert S.(1952). *Jean Piaget: A History of Psychology in Autobiography*, Vol. IV, pp. 237–256, doi: 10.1037/11154-011, Worcester: Clark University Press.

Bovet Magali (1976). *Piaget's Theory of Cognitive Development and Individual Differences, Piaget and His School*, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 269–279, doi:10.1007/978-3-642-46323-5_20.

Bruner J. S. (1978). "Learning how to do things with words", in: J. S. Bruner and R. A. Garton (Eds.), *Human Growth and Development*, Oxford: Clarendon Press, pp. 62–84.

Bruner J. S. (1983). "The acquisition of pragmatic commitments", in: R. Golinkoff (Ed.), *The Transition From Prelinguistic to Linguistic Communication*, Hillsdale, NJ: Lawrence Erlbaum Associates, pp. 27–42.

Bruner J. (1995). "From joint attention to the meeting of minds", in: C. Moore & P. Dunham (Eds.), *Joint Attention: Its Origins and Role in Development*, Hillsdale, N.J.: Erlbaum.

Cherry K. (March 31, 2020). "What are Piaget's four stages of development?", *Verywell Mind*, available online at: https://www.verywellmind.com/piagets-stages-of-cognitive-development-2795457.

Chen L., Chang Y. and Yan L. (2022). "On-orbit real-time variational image destriping: FPGA architecture and implementation", *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 60, doi: 10.1109/tgrs.2022.3140428.

Daher W. and Awawdeh Shahbari J. (2020). "Design of STEM activities: Experiences and perceptions of prospective secondary school teachers", *International Journal of Emerging Technologies in Learning (iJET)*, Vol. 15, No. 4, pp. 112–128.

Du C. and Yamaguchi Y. (2020). "High-level synthesis design for stencil computations on FPGA with high bandwidth memory", *Electronics*, Vol. 9, No. 8, p. 1275, doi: https://doi.org/10.3390/electronics9081275.

Erotokritou Stavrou Th. and Koutselini M. (2016). "Differentiation of teaching and learning: The teachers' perspective", *Universal Journal of Educational Research*, Vol. 4, No. 11, pp. 2581–2588, doi: https://doi.org/10.13189/ujer.2016.041111.

Faber J. M., Glas C. A. W. and Visscher A. J. (2018). "Differentiated instruction in a data-based decision-making context", *School Effectiveness and School Improvement*, Vol. 29, pp. 46–63, doi: https://doi.org/10.1080/09243453.2017.1366342.

Gagne R. M., Wager W. W., Golas K. C., Keller J. M. and Russell J. D. (2005). "Principles of instructional design", *Performance Improvement*, Vol. 44, pp. 44–46, doi: https://doi.org/10.1002/pfi.4140440211.

Gujar A. (2014). "Image encryption using AES algorithm based on FPGA", *International Journal of Computer Science and Information Technologies*, Vol. 5, No. 5, pp. 6853–6859.

Handrianto, C., & Rahman, M. A. (2018). Project-based learning: a review of literature on its outcomes and implementation issues. LET Linguistics, Literature, and English Teaching Journal, 8(2), 110-129.

Hao X., Lin C. and Wu Q. (2020). "A parallel timing synchronization structure in real-time high transmission capacity wireless communication systems", *Electronics*, Vol. 9, No. 4, p. 652, doi: https://doi.org/10.3390/electronics9040652.

Jewitt C. (2008). "Multimodality and literacy in school classrooms", *Review of Research in Education*, Vol. 32, pp. 241–267.

Joel Dunham and Eric Johnson (15 Jun., 2012). "Integrated Pan/Tilt sensor system and flight controls", *AIAA Guidance, Navigation and Control Conference and Exhibit*, doi: https://doi.org/10.2514/6.2008-7316.

Kennedy T. J. and Odell M. R. L. (2014). "Engaging students in STEM education", *Science Education International*, Vol. 25, No. 3, pp. 246–258.

Kurt S. (August 8, 2020). "Jean Piaget and his theory & stages of cognitive development", *Educational Technology*, available online at: https://educationaltechnology.net/jean-piaget-and-his-theory-stages-of-cognitive-development/.

Muthukrishnan S. and Priyadharsini R. (Dec. 2014). "32-bit RISC and DSP system design in an FPGA", *International Journal of Computer Science and Mobile Computing*, Vol. 3, No. 12, pp. 361–368.

Parikh R. S. (Jan. 2018). "Alarm system implementation on field programmable gate array", *Int. Journal of Engineering Research and Application*, Vol. 8, No. 1, pp. 1-4.

Sanjay Singh, Chandra Shekhar and Anil Vohra (2017). "Real-time FPGA-based object tracker with automatic pan-tilt features for smart video surveillance systems", *Journal of Imaging*, Vol. 3, No. 2, p. 18, doi: https://doi.org/10.3390/jimaging3020018.

Saroch K. and Sharma A. (Mar.-Apr. 2013). "FPGA based system login security lock design using finite state machine", *IOSR Journal of Electronics and Communication Engineering*, Vol. 5, No. 3, pp. 70–75.

Sebastian Strube (August 2011). "Compact field programmable gate array (FPGA)-based multi-axial interferometer for simultaneous tilt and distance measurement in the sub-nanometre range", *Gabor Molnar and Hans-Ulrich Danzebrink, Measurement Science and Technology*, Vol. 22, Doi: 10.1088/0957-0233/22/9/094026.

Singh S., Saini A. K., Saini R., Image I. J. (2014). "Interfacing the Analog camera with FPGA board for real-time video acquisition", *Graphics and Signal Processing*, No. 4, pp. 32–38, doi: 10.5815/ijigsp.2014.04.04.

Yarlagadda S., Kaza S., Tummala A., Babu E. and Prabhakar R. (2021). "The reduction of Crosstalk in VLSI due to parallel bus structure using Data Compression Bus Encoding technique implemented on Artix 7 FPGA Architecture", *Information Technology in Industry*, Vol. 9, No. 1, pp. 456–460, doi: 10.17762/itii.v9i1.151, 2021.

Yussup M., Ibrahim M. M., Lombigit L., Rahman N. A. A. and Zin M. R. M. (2014). "Implementation of data acquisition interface using on-board field-programmable gate array (FPGA) universal serial bus (USB) link", *Advance. in Nuclear Research and Energy Development, AIP Conf. Proc. 1584*, pp. 69–72, doi: 10.1063/1.4866106.